

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH



BÁO CÁO MÔN HỌC

**ĐỀ TÀI: THIẾT KẾ HỆ THỐNG IOT TƯỚI NƯỚC TỰ ĐỘNG SỬ DỤNG TRONG
NÔNG NGHIỆP**

GIẢNG VIÊN: Ths HUỲNH VĂN HÓA

Môn: Phát triển hệ thống và ứng dụng IoT

Sinh viên :

Nguyễn Phước An	N19DCVT001
Võ Trung Tín	N19DCVT055
Chu Đình Huấn	N19DCVT013
Đoàn Nam Như Thiện	N19DCVT063

Năm học: 2023-2024

MỞ ĐẦU

Ngày nay việc phát triển ngành nông nghiệp đang là một hướng đi mang lại nguồn lợi kinh tế cho nước nhà. Nhưng nếu chỉ áp dụng phương pháp nông nghiệp truyền thống sẽ mang lại hiệu quả kinh tế thấp. Chính vì vậy việc áp dụng công nghệ kỹ thuật vào nông nghiệp là một hướng đi thông minh mang lại những giá trị bền vững.

Trong đó, ứng dụng Internet of Things (IoT) về mô hình hệ thống tự động thông qua cảm biến cung cấp chính xác các thông tin như độ ẩm đất, nhiệt độ, ánh sáng, ...đồng bộ hóa dữ liệu lên website từ đó điều tiết việc tưới cây sao cho hợp lý giúp chúng ta giảm áp lực trong việc chăm sóc cây trồng, chi phí nhân công và các nhân tố xấu ảnh hưởng đến cây trồng.

Vì vậy, nhóm em quyết định thực hiện nghiên cứu đề tài “Thiết kế hệ thống tưới cây tự động – Lập trình Arduino và ESP8266” với mục đích tạo ra một hệ thống chăm sóc cây trồng có giá thành hợp lý, hiệu quả quan trọng hơn là có thể áp dụng rộng rãi trong nông nghiệp nước nhà.

LỜI CẢM ƠN

Với lòng biết ơn chân thành, nhóm em muốn bày tỏ sự tôn trọng và cảm kích đến thầy Ths. Huỳnh Văn Hóa trong môn học Phát triển hệ thống và ứng dụng IoT. Dưới sự hướng dẫn tận tình và nhiệt tình của thầy, nhóm em đã hoàn thành đề tài thành công, và đã có được kiến thức, kinh nghiệm cần thiết để phát triển trong tương lai.

Chúng em chắc chắn sẽ không quên được sự giúp đỡ của thầy, từ những lời khuyên, định hướng đến những bài giảng chất lượng cao. Những kỹ năng và kinh nghiệm mà thầy truyền đạt sẽ luôn là tài sản vô giá trong cuộc đời học tập và sự nghiệp của chúng em.

Nhóm em xin chân thành cảm ơn thầy Huỳnh Văn Hóa vì những giờ phút quý giá đã dành cho chúng em. Thầy là một người thầy tuyệt vời và chúng em rất tự hào khi có cơ hội được học tập dưới sự chỉ bảo của thầy. Chúc thầy sức khỏe, hạnh phúc và thành công trong sự nghiệp giáo dục.

MỤC LỤC

MỞ ĐẦU.....	2
LỜI CẢM ƠN	3
MỤC LỤC	4
DANH MỤC HÌNH	7
CHƯƠNG I: GIỚI THIỆU.....	8
1. Lý do nghiên cứu.....	8
2. Lịch sử nghiên cứu.....	8
3. Mục tiêu nghiên cứu.	8
3.1. Mục đích.....	8
3.2. Nhiệm vụ.	9
4. Phạm vi nghiên cứu.	9
5. Vấn đề nghiên cứu.	9
6. Kinh phí thực hiện đề tài.....	9
CHƯƠNG II : Ý TƯỞNG THIẾT KẾ.....	12
1. Chức năng.....	12
2. Nguyên lí hoạt động	12
CHƯƠNG III: THỰC HIỆN ĐỀ TÀI.....	13
1. Phần cứng	13
1.1 Arduino Uno R3.	13
1.2 Cảm biến đo nhiệt độ ngoài trời LM35.....	14

1.3	Mạch chuyển đổi I2C cho LCD	15
1.4	Cảm biến đo độ ẩm trong đất	16
1.5	Module 1 Relay Với Opto Cách Ly Kích H/L (5VDC).....	17
1.6	Màn hình LCD 1602	18
1.7	Động cơ bơm 12V	19
1.8	Điện trở và Led.....	19
1.9	Đèn 220V.	20
2.	PHẦN MỀM MÔ PHỎNG PROTEUS	22
3.	ARDUINO IDE VÀ LẬP TRÌNH CHO ARDUINO.....	23
3.1	Mô phỏng mạch nguyên lý trên Proteus.....	24
3.2	Nguyên lý hoạt động:.....	24
4.	LƯU ĐỒ THUẬT TOÁN	25
5.	SƠ ĐỒ LẮP ĐẶT	25
5.1:	KẾT NỐI GIỮA ARDUINO VÀ CÁC THIẾT BỊ.	25
5.2:	KẾT NỐI GIỮA ARDUINO VÀ ESP8266.....	26
6	THIẾT KẾ APP ĐIỀU KHIỂN QUA ĐIỆN THOẠI.	27
6.1	Giới thiệu về Blynk.....	27
6.2.	Cấu hình và mô tả cách hoạt động của Blynk IoT.	28
7.	CHƯƠNG TRÌNH CHO ARNODEMCU.....	30
8.	CHƯƠNG TRÌNH CHO ESP8266.....	36
8.1	Chú thích	39
CHƯƠNG IV : KẾT QUẢ ĐẠT ĐƯỢC		40
1.	Trường hợp 1.....	40
2.	Trường hợp 2.....	40

3. Trường hợp 3.....	41
CHƯƠNG V : ĐỀ XUẤT PHÁT TRIỂN ĐỀ TÀI.....	43
TÀI LIỆU THAM KHẢO	44

DANH MỤC HÌNH

Hình 1: Arduino Uno R3.....	13
Hình 2: LM35 - Temperature Sensor	14
Hình 3: Module I2C	15
Hình 4: Module cảm biến độ ẩm đất	16
Hình 5: Module relay 5V	17
Hình 6: Màn hình LCD 1602.....	18
Hình 7. Động cơ bơm 12V.....	19
Hình 8: Điện trở 220 Ω	20
Hình 9: Led	20
Hình 10: Đèn 220v	20
Hình 11: NodeMCU ESP8266	21
Hình 12: Giao diện phần mềm Proteus	23
Hình 13: Giao diện phần mềm Arduino IDE	23
Hình 14: Mạch nguyên lý vẽ trên Proteus.....	24
Hình 15: Lưu đồ thuật toán.....	25
Hình 16: Sơ đồ lắp đặt	26
Hình 17: Sơ đồ lắp đặt	26
Hình 18: Hình ảnh app blynk đã được cài đặt.	28
Hình 19: Trường hợp 1 thiếu độ ẩm motor hoạt động	40
Hình 20: Trường hợp 2 đủ độ ẩm motor tắt đèn 220v hoạt động.....	41
Hình 21: Trường hợp 3 dư độ ẩm motor tắt đèn 220v sáng.....	42

CHƯƠNG I: GIỚI THIỆU

1. Lý do nghiên cứu.

Ngày nay, nước ta đang hướng tới xây dựng một nền công nghiệp phát triển, điều đó mang lại nhiều lợi ích cho đất nước nhưng cũng kèm theo nhiều hệ lụy, đặc biệt là ô nhiễm môi trường, một trong những mảng chịu thiệt hại rất lớn từ vấn đề trên chính là ngành nông nghiệp. Bên cạnh đó, trong nhiều năm qua, sản xuất nông nghiệp ở Việt Nam gặp rất nhiều khó khăn do các hiện tượng thời tiết cực đoan ngày càng diễn biến phức tạp do ảnh hưởng của biến đổi khí hậu toàn cầu, làm ảnh hưởng to lớn đến năng suất cũng như chất lượng các sản phẩm từ nông nghiệp.

Một trong những ứng dụng công nghệ nổi bật được đưa vào trong nông nghiệp trong những năm gần đây là Internet of Things (viết tắt là IOT) đã và đang đem lại nhiều kết quả thành công, dần dần được áp dụng và phổ biến trên nhiều diện tích canh tác nông nghiệp, vì vậy chúng em chọn đề tài “Thiết kế hệ thống IoT tưới nước tự động sử dụng trong nông nghiệp” nhằm có hiểu biết thêm về tác động của công nghệ tới khả năng phát triển của cây trồng, cũng như quản lý của người điều khiển, bên cạnh đó là nghiên cứu thêm về các ứng dụng công nghệ điện tử được đưa vào.

2. Lịch sử nghiên cứu.

- (1) Lê Trường Thuận, Phan Thành Nam, Khoa Công Nghệ, Trường Đại Học Cần Thơ, “*Thiết kế hệ thống tưới nước tự động sử dụng arduino*”, được viết vào năm 2019. Bài viết đề cập đến việc ngoài việc sử dụng các phương pháp tưới cây phổ thông, chúng ta có thể tạo ra các phương pháp tưới cây tự động đơn giản cho khu vườn nhỏ của mình thậm chí có thể mở rộng hệ thống tưới cây cho cả một khu vườn lớn.
- (2) Phạm Minh Anh, Khoa Điện Tử Tự Động Công Nghiệp, Trường Đại Học Dân Lập Hải Phòng, “*Thiết kế hệ thống tưới nước tự động*”, được viết vào năm 2016. Bài viết đề cập đến việc trong cuộc sống hằng ngày, nhiều người có thú vui trồng những cây cảnh, vườn rau... nhưng lại bận rộn ở các công việc hằng ngày nên không ai tưới nước cho cây, nên đã thiết kế nên đề tài “Thiết kế hệ thống tưới nước tự động”, để thay con người tưới nước hằng ngày.

3. Mục tiêu nghiên cứu.

3.1. Mục đích

Mục đích của đề tài là nghiên cứu, thiết kế và triển khai một hệ thống tưới cây tự động trong lĩnh vực nông nghiệp, sử dụng Arduino hoặc các công nghệ tương tự, nhằm cải thiện hiệu quả và tiết kiệm tài nguyên trong quá trình tưới cây. Đồng thời, đề tài cũng nhằm đánh giá tính hiệu quả và độ chính xác của hệ thống thông qua các thử nghiệm và phân tích dữ liệu. Ngoài ra, đề tài có thể đề xuất các cải tiến và phát triển để nâng cao tính ổn định, độ tin cậy, và khả năng ứng dụng thực tế của hệ thống tưới cây tự động trong lĩnh vực nông nghiệp.

3.2. Nhiệm vụ.

- (1) Tìm ra các phương pháp, công nghệ và thiết bị được sử dụng trong hệ thống tưới cây tự động sử dụng trong nông nghiệp.
- (2) Xác định tính hiệu quả và độ chính xác của hệ thống tưới nước tự động sử dụng trong nông nghiệp.
- (3) Đề xuất ra các giải pháp cải tiến và phát triển để hoàn thiện hệ thống tưới nước tự động sử dụng trong nông nghiệp.

4. Phạm vi nghiên cứu.

4.1. Đối tượng nghiên cứu: Hệ thống tưới cây tự động sử dụng.

4.2. Khách thể nghiên cứu: Nông dân và người quản lý cây trồng .

4.3. Phạm vi nghiên cứu: Trong nông nghiệp.

5. Vấn đề nghiên cứu.

- 5.1.** Các phương pháp , công nghệ và thiết bị được sử dụng trong đề tài gồm những gì?
- 5.2.** Làm thế nào để đánh giá tính hiệu quả và độ chính xác của hệ thống tưới nước tự động sử dụng trong nông nghiệp?
- 5.3.** Các giải pháp cải tiến và phát triển để hoàn thiện hệ thống là gì?

6. Kinh phí thực hiện đề tài.

Các thiết bị	Giá
1.Đế test bread board MB-102	24.000đ
2.Led màu đỏ (10 con)	2.000đ
3.LCD 1602 xanh lá 5V	40.000đ
4.Adapter 12V2A (Cục nguồn motor)	50.000đ
5.Dây bus, dây bẹ 2 đầu đực 20cm (10 sợi)	16.000đ
6.Dây bus, dây bẹ đực cái 20cm (10 sợi)	16.000đ
7.Module I2C giao tiếp LCD1602...	19.000đ
8.Module relay 5V (2 con)	32.000đ
9.Cảm biến độ ẩm đất	12.000đ
10.Cảm biến nhiệt độ (LM35)	35.000đ
11.Ống bơm nước 6MM	7.000đ
12.Bơm nước 365DC 12V	50.000đ
13.Module mở rộng chân PCF8574T giao tiếp I2C	29.000đ
14. Arduino uno R3	190.000đ
15. Bóng đèn trái ốt 220V	8.000đ
16.Phích cắm tròn	8.000đ
17.Dây rút 3x100mm	5.000đ
18.Module ESP8266	83.000đ

19.LM35	32.000đ
Tổng tiền	668.000đ

CHƯƠNG II : Ý TƯỞNG THIẾT KẾ

1. Chức năng

“Hệ thống tưới cây tự động” đáp ứng được các yêu cầu chức năng sau:

- Đọc nhiệt độ của không khí và độ ẩm đất thông qua cảm biến.
- Hiển thị thông số đọc được từ cảm biến lên màn hình LCD .
- Tự động bật, tắt máy bơm sau khi xử lý độ ẩm đo được bằng module relay.
- Điều khiển đèn 220V
- Sử dụng app Blynk IoT để điều khiển và quan sát hoạt động của hệ thống trên điện thoại.

2. Nguyên lí hoạt động

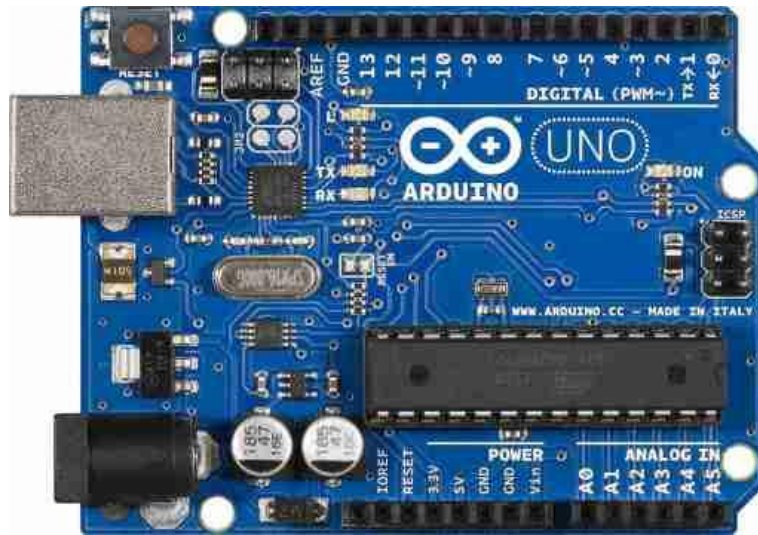
“Hệ thống tưới nước tự động” sẽ hoạt động dựa trên thiết bị cảm biến độ ẩm đất và cảm biến nhiệt độ . Thông số về nhiệt độ, độ ẩm đọc được từ cảm biến sẽ hiển thị lên màn hình LCD. Các thông số đọc được sẽ được xử lý bằng thuật toán thông qua mã code được nạp vào module, khi thông số nhiệt độ hay độ ẩm dưới mức cho phép đã cài đặt sẽ kích hoạt relay để máy bơm hoạt động và bật đèn và ngược lại .

CHƯƠNG III: THỰC HIỆN ĐỀ TÀI

1. Phần cứng

1.1 Arduino Uno R3.

Khi nhắc tới dòng mạch Arduino dùng để lập trình hay nghiên cứu chế tạo thì dòng đầu tiên mà người ta thường tìm hiểu đó chính là dòng Arduino UNO. Hiện dòng mạch này đã phát triển tới thế hệ thứ 3 (R3). Nếu bạn là người mới tìm hiểu thì tôi khuyên bạn nên sử dụng Arduino Uno R3 để phục vụ cho việc học tập và nghiên cứu của mình hơn là các dòng Arduino khác vì dòng Arduino Uno R3 có kích thước nhỏ gọn, đầy đủ tính năng và rất dễ sử dụng đối với những người mới tiếp cận về lập trình.



Hình 1: Arduino Uno R3

Vi điều khiển	Atmega328 (8bit)
Điện áp hoạt động	5V DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16MHz
Dòng tiêu thụ	Khoảng 30mA
Điện áp vào khuyến dùng	7-12V DC
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	14 (có 6 chân PWM)

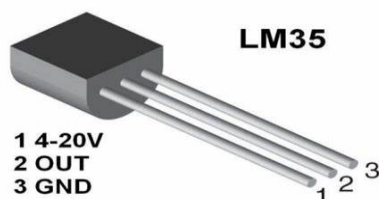
Số chân Analog	6 (độ phân giải 10bit)
Dòng ra tối đa (5V)	500mA
Dòng ra tối đa (3.3V)	50mA
Bộ nhớ flash	32 KB (ATmega328) với 0.5KB dùng bởi bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Dòng tối đa trên mỗi chân I/O	30 mA

Thông số kỹ thuật: Arduino UNO R3

1.2 Cảm biến đo nhiệt độ ngoài trời LM35

Cảm biến nhiệt độ LM35 là một cảm biến nhiệt độ analog được sử dụng để đo nhiệt độ với độ chính xác cao. Cảm biến này được thiết kế để hoạt động ở dải nhiệt độ từ -55 đến +150 độ C, và có độ chính xác tuyệt đối là $\pm 0,5$ độ C ở nhiệt độ phòng.

LM35 có đầu ra analog, nghĩa là nó sẽ tạo ra một điện áp đầu ra tương ứng với nhiệt độ được đo. Điện áp đầu ra tăng 10mV cho mỗi độ C tăng nhiệt độ, điều này rất thuận tiện để xử lý tín hiệu. Ngoài ra, LM35 có thể được cấp nguồn từ nguồn 5V DC thông thường và không cần thêm mạch đo đặc nhiệt độ. Cảm biến LM35 rất thích hợp cho các ứng dụng cần độ chính xác cao trong đo lường nhiệt độ như trong các hệ thống điều khiển nhiệt độ, thiết bị y tế và các ứng dụng khoa học.



Hình 2: LM35 - Temperature Sensor

Số chân	Tên chân	Chức năng
1	V _{CC} hay +V _S	Chân cấp nguồn với điện áp từ 4V đến 30V
2	V _{OUT}	Chân lấy điện áp ra, điện áp ở chân này thay đổi 10mV/°C
3	GND	Chân nối đất

Thông số kỹ thuật : LM35

1.3 Mạch chuyển đổi I2C cho LCD

LCD có quá nhiều chân gây khó khăn trong quá trình kết nối và chiếm dụng nhiều chân của vi điều khiển? Module chuyển đổi I2C cho LCD sẽ giải quyết vấn đề này cho bạn, thay vì sử dụng tối thiểu 6 chân của vi điều khiển để kết nối với LCD (RS, EN, D7, D6, D5 và D4) thì với module chuyển đổi bạn chỉ cần sử dụng 2 chân (SCL, SDA) để kết nối. Module chuyển đổi I2C hỗ trợ các loại LCD sử dụng driver HD44780(LCD 1602, LCD 2004).



Hình 3: Module I2C

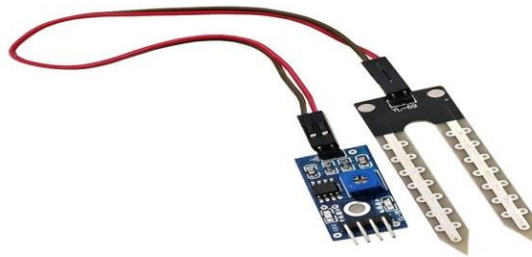
Điện áp hoạt động	2.5-6V DC
Hỗ trợ màn hình	LCD1602,1604,2004 (driver HD44780)
Giao tiếp:	I2C
Địa chỉ mặc định	0x27 (có thể điều chỉnh bằng ngắn mạch chân A0/A1/A2)
Kích thước	41.5mm(L)x19mm(W)x15.3mm(H)
Trọng lượng	5g

Tích hợp Jump chốt để cung cấp đèn cho LCD hoặc ngắt
Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD

Thông số kỹ thuật : I2C

1.4 Cảm biến đo độ ẩm trong đất

Cảm biến đo độ ẩm đất là một module cảm biến độ ẩm dùng để đóng cắt relay khi độ ẩm của đất ở nơi đo thay đổi quá ngưỡng chúng ta cài đặt. Minh thường đặt đầu ra của module sẽ ở mức thấp, khi cảm biến phát hiện thiếu nước. Module sẽ chuyển về mức cao, điều khiển relay đóng và máy bơm hoạt động. Khi nước đã được bơm đầy, cảm biến phát hiện đủ nước. Module tự động về mức thấp, điều khiển mở relay.



Hình 4: Module cảm biến độ ẩm đất

Điện áp hoạt động	3.3V-5V
Kích thước PCB	3cm * 1.6cm
IC so sánh	LM393
VCC	3.3V-5V
GND	0V
D0	Đầu ra tín hiệu số (0 và 1)
A0	Đầu ra Analog (Tín hiệu tương tự).

Led đỏ báo nguồn vào, Led xanh báo độ ẩm.

Thông số kỹ thuật : module cảm biến độ ẩm đất.

1.5 Module 1 Relay Với Opto Cách Ly Kích H/L (5VDC)

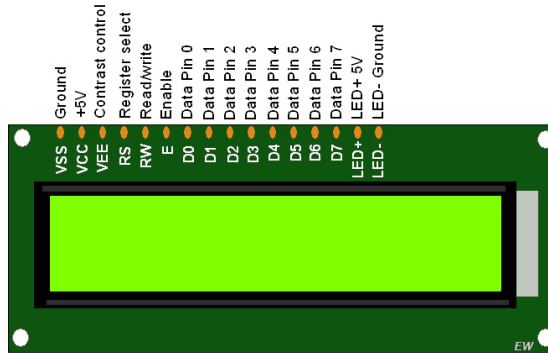
Module 1 Relay với opto cách ly nhỏ gọn, có opto và transistor cách ly giúp cho việc sử dụng trở nên an toàn với board mạch chính, module 1 Relay với opto cách ly 5v được sử dụng để đóng ngắt nguồn điện công suất cao AC hoặc DC, có thể chọn đóng khi kích mức cao hoặc mức thấp bằng Jumper. Tiếp điểm đóng ngắt gồm 3 tiếp điểm NC (thường đóng), NO(thường mở) và COM(chân chung) được cách ly hoàn toàn với board mạch chính, ở trạng thái bình thường chưa kích NC sẽ nối với COM, khi có trạng thái kích COM sẽ chuyển sang nối với NO và mất kết nối với NC.



Hình 5: Module relay 5V

Điện thế đóng ngắt tối đa	AC250V ~ 10A hoặc DC30V ~ 10A
Kích thước	1.97 in x 1.02 in x 0.75 in (5.0 cm x 2.6 cm x 1.9 cm)
Weight	0.60oz (17g)
Sử dụng điện áp nuôi DC 5V	
Relay mỗi Relay tiêu thụ dòng khoảng 80mA	
Có đèn báo đóng ngắt trên mỗi Relay.	
Có thể chọn mức tín hiệu kích 0 hoặc 1 qua jumper	

1.6 Màn hình LCD 1602



Hình 6: Màn hình LCD 1602.

Màn hình LCD 1602 xanh lá có khả năng hiển thị 2 dòng với mỗi dòng 16 ký tự, màn hình có độ bền cao, rất phổ biến và dễ dàng sử dụng hơn nếu đi kèm mạch chuyển tiếp I2C.

Điện áp MAX	7V
Điện áp MIN	-0,3V
Hoạt động ổn định	2.7-5.5V
Điện áp ra mức cao	>2.4V
Điện áp ra mức thấp	<0.4V
Nhiệt độ hoạt động	-30 đến 75 độ C
Chữ trắng , nền xanh	
Khoảng cách giữa hai chân kết nối là 0.1 inch tiện dụng khi kết nối với Breadboard.	
Tên các chân được ghi ở mặt sau của màn hình LCD hỗ trợ việc kết nối, đi dây điện.	
Có đèn led nền, có thể dùng biến trở hoặc PWM điều chỉnh độ sáng để sử dụng ít điện năng hơn.	

Thông số kỹ thuật :LCD 1602

1.7 Động cơ bơm 12V



Hình 7. Động cơ bơm 12V

Động cơ bơm 12V là một loại động cơ điện được thiết kế để hoạt động ở điện áp 12V DC. Nó thường được sử dụng trong các ứng dụng bơm nước nhỏ, như bơm nước trong xe hơi, tàu thuyền hoặc máy móc xây dựng. Động cơ bơm 12V thường có kích thước nhỏ gọn, có thể hoạt động ở mức tiêu thụ điện năng thấp, giúp tiết kiệm năng lượng và làm giảm chi phí hoạt động

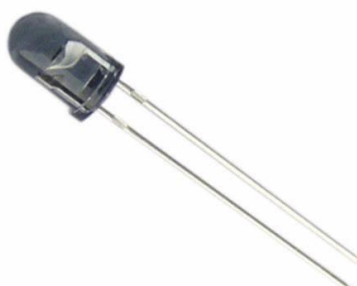
Loại động cơ	P385
Điện áp sử dụng	6~12VDC
Dòng điện sử dụng	0.5~0.7A
Lưu lượng bơm	1~2L / 1 phút
Thời gian chạy liên tục	< 1h
Đường kính đường bơm	ĐK trong 6mm , ĐK ngoài 8.5mm
Kích thước	90 x 40 x 35 mm

Thông số kỹ thuật: motor

1.8 Điện trở và Led.



Hình 8: Điện trở 220 Ω



Hình 9: Led

1.9 Đèn 220V.

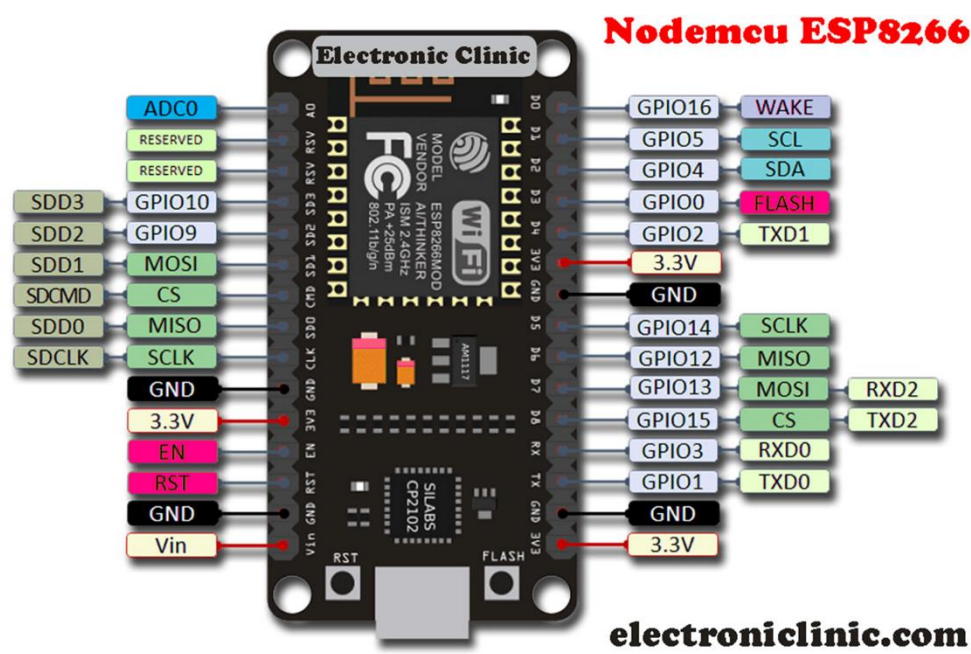


Hình 10: Đèn 220v

Điện áp	220V AC
Đuôi vặn ren	12mm (E12)
Công suất	0.3W
Bóng led trái ốt	

Thông số kỹ thuật: Đèn 220V

1.10:ModuleMCU ESP8266.



Hình 11: NodeMCU ESP8266

Vi điều khiển	Atmega328 (8bit)
Điện áp hoạt động	3.3V DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16MHz
Dòng tiêu thụ	Khoảng 30mA
Điện áp vào khuyến dùng	5V thông qua cổng USB
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)
Số chân Analog	1(điện áp vào tối đa 3.3V)
Dòng ra tối đa (5V)	500mA

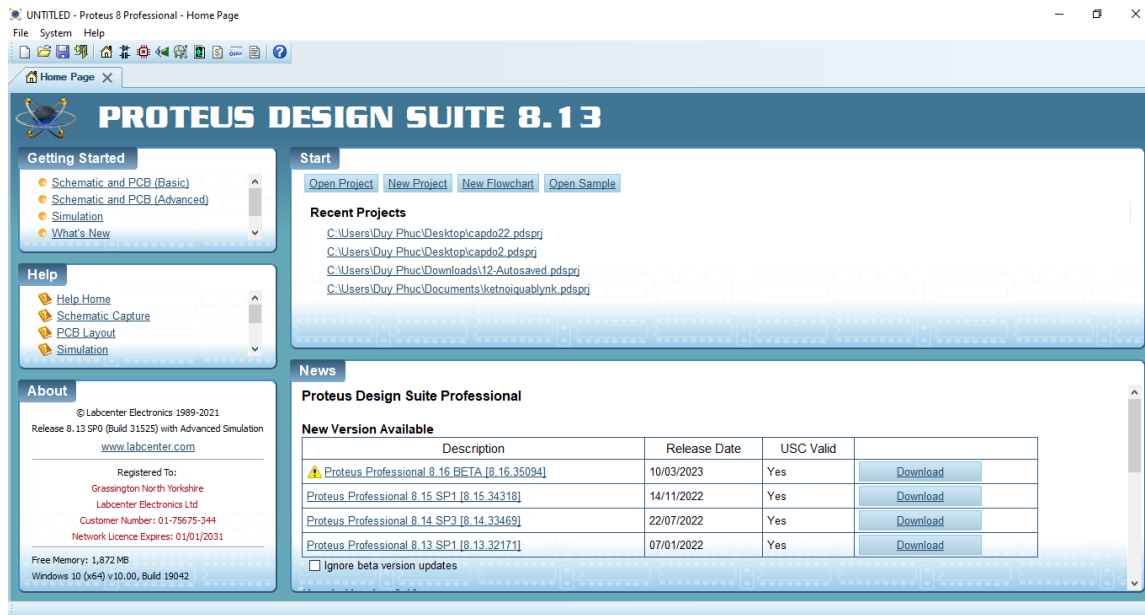
Dòng ra tối đa (3.3V)	50mA
Bộ nhớ flash	32 KB (ATmega328) với 0.5KB dùng bởi bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Dòng tối đa trên mỗi chân I/O	30 mA

Thông số kỹ thuật: *NodeMCU ESP8266*

2. PHẦN MỀM MÔ PHỎNG PROTEUS

Phần mềm Proteus là phần mềm cho phép mô phỏng hoạt động của mạch điện tử bao gồm phần thiết kế mạch và viết chương trình điều khiển cho các họ vi điều khiển MCS-51, PIC, AVR,... Proteus là phần mềm mô phỏng mạch điện tử của Labcenter Electronic, mô phỏng cho hầu hết các linh kiện điện tử thông dụng, đặc biệt hỗ trợ cho các MCU như PIC, 8051, AVR, Motorola.

Phần mềm bao gồm 2 chương trình: ISIS cho phép mô phỏng mạch và ARES dùng để vẽ mạch in. Proteus là phần mềm mô phỏng cho các loại vi điều khiển khá tốt, hỗ trợ các dòng vi điều khiển PIC, 8051, dsPIC, AVR, HC11,... các giao tiếp I2C, SPI, CAN, USB, Ethernet,... ngoài ra còn mô phỏng các mạch số, mạch tương tự một cách hiệu quả .

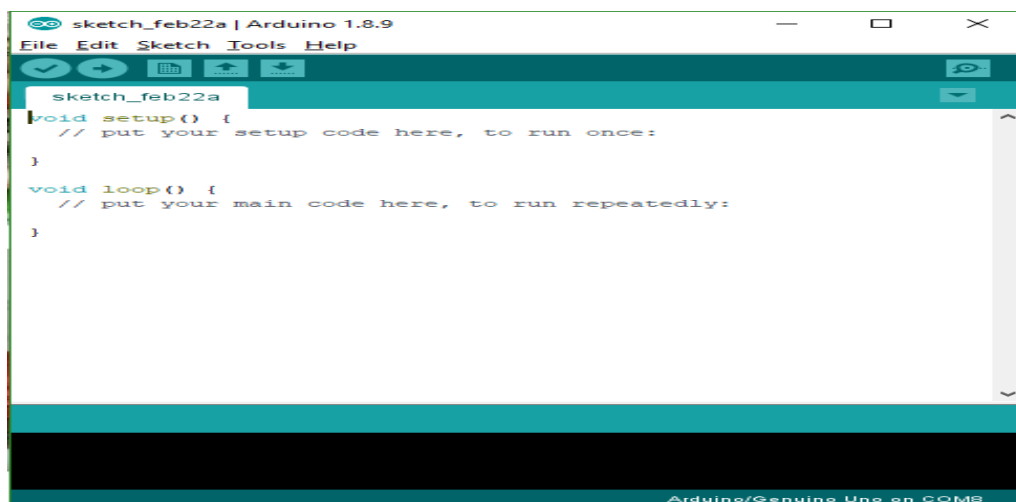


Hình 12: Giao diện phần mềm Proteus

3. ARDUINO IDE VÀ LẬP TRÌNH CHO ARDUINO

Arduino IDE được viết tắt (Arduino Integrated Development Environment) là một trình soạn thảo văn bản, giúp bạn viết code để nạp vào bo mạch Arduino.

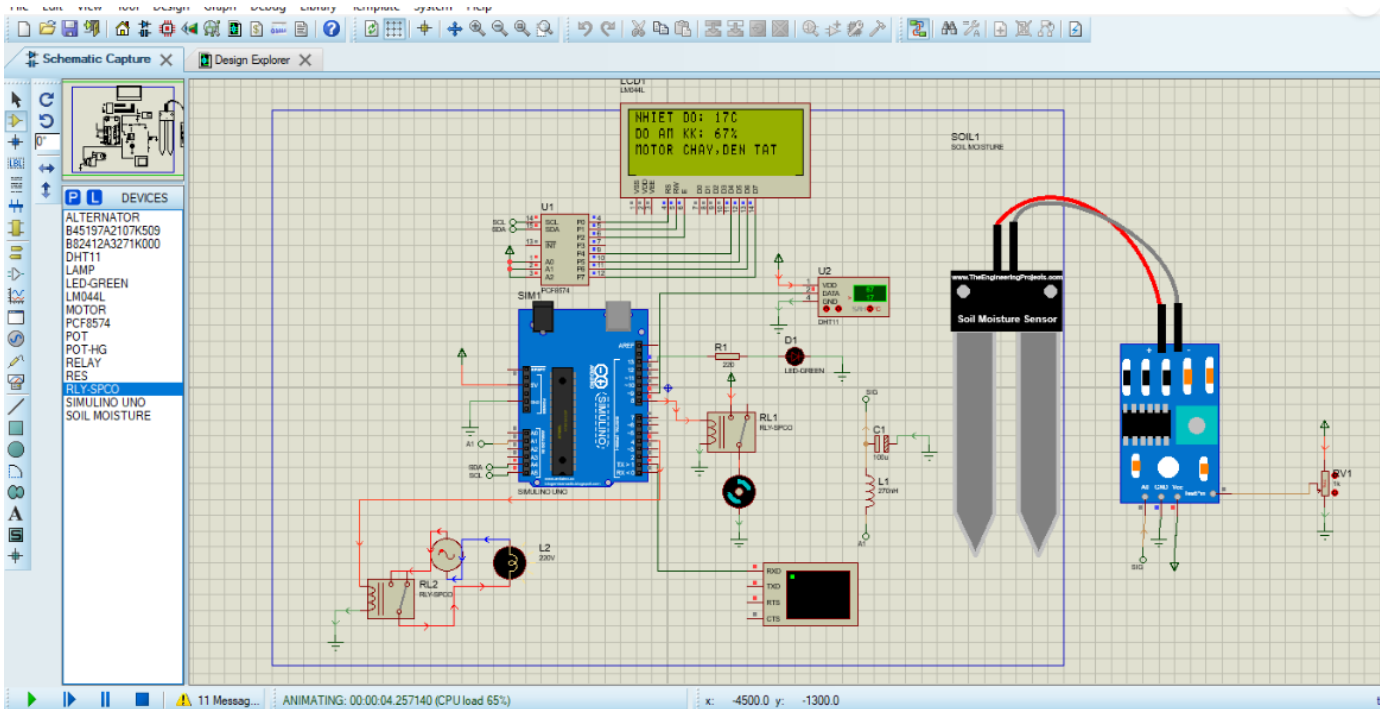
Thiết kế bo mạch nhỏ gọn, trang bị nhiều tính năng thông dụng mang lại nhiều lợi thế cho Arduino, tuy nhiên sức mạnh thực sự của Arduino nằm ở phần mềm. Môi trường lập trình đơn giản dễ sử dụng, ngôn ngữ lập trình Wiring dễ hiểu và dựa trên nền tảng C/C++ rất quen thuộc với người làm kỹ thuật. Và quan trọng hơn là số lượng thư viện code được viết sẵn và chia sẻ bởi cộng đồng nguồn mở là cực kỳ lớn.



Hình 13: Giao diện phần mềm Arduino IDE

Arduino IDE là phần mềm dùng để lập trình cho Arduino. Môi trường lập trình cho Arduino là IDE có thể chạy trên ba nền tảng phổ biến nhất hiện nay là Windows, Macintosh OSX là Linux. Do có tính chất nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng bởi người dùng có kinh nghiệm .

3.1 Mô phỏng mạch nguyên lý trên Proteus



Hình 14: Mạch nguyên lý vẽ trên Proteus

3.2 Nguyên lý hoạt động:

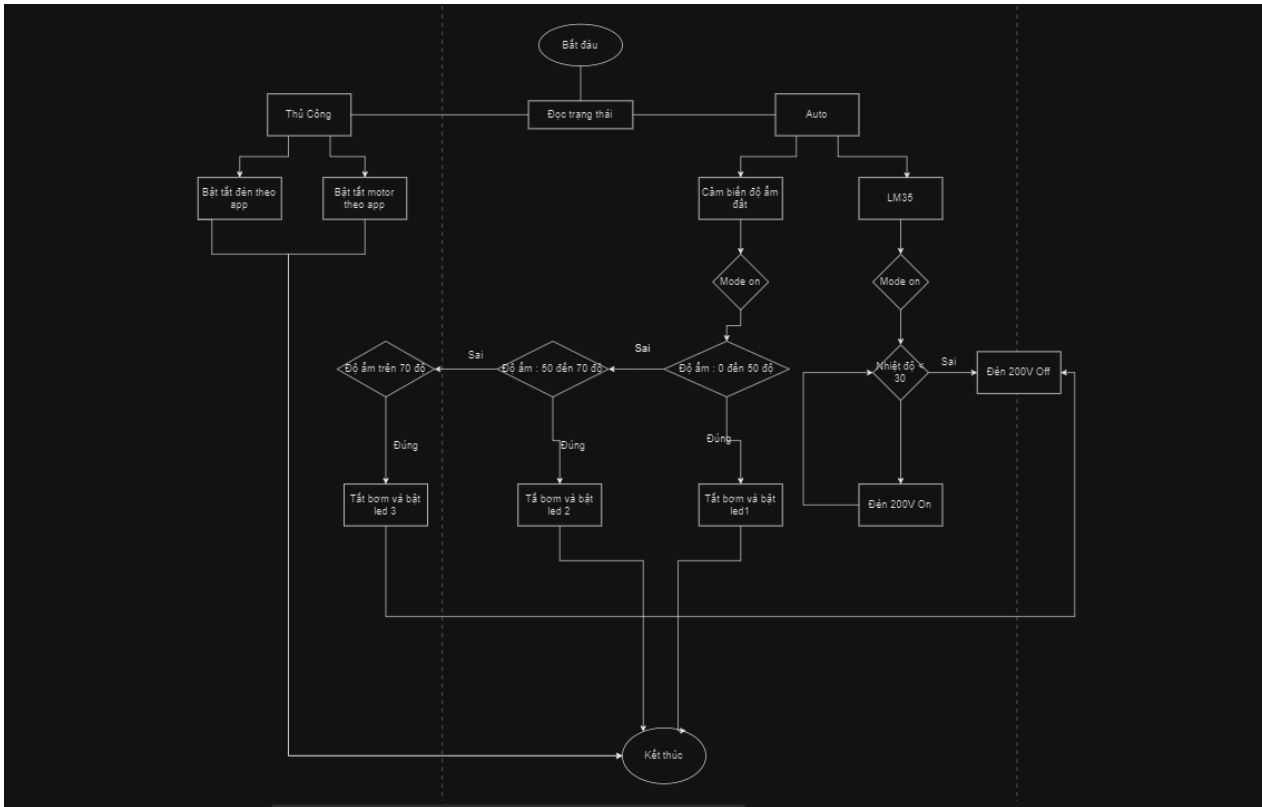
Giá trị độ ẩm của đất được nhận biết qua cảm biến độ ẩm và đưa về bộ điều khiển là Arduino UNO R3 bằng tín hiệu Analog. Khi đất không đủ độ ẩm, giá trị tín hiệu Analog đọc được từ cảm biến sẽ thấp hơn một ngưỡng được xác định trước đó. Bộ điều khiển sẽ đưa tín hiệu cho đóng Relay và động cơ máy bơm bắt đầu hoạt động bơm nước tưới vào chậu. Khi đã đủ nước, độ ẩm cao, giá trị tín hiệu Analog đọc được từ cảm biến sẽ vượt qua ngưỡng trước đó. Bộ điều khiển sẽ đưa tín hiệu ngắt Relay, và máy bơm sẽ dừng hoạt động.

Led xanh dương có tác dụng báo hiệu độ ẩm thấp, cây đang thiếu nước.

Led vàng có tác dụng báo hiệu đủ độ ẩm ,cây đã đủ nước.

Led đỏ có tác dụng báo dư độ ẩm ,cây thừa nước.

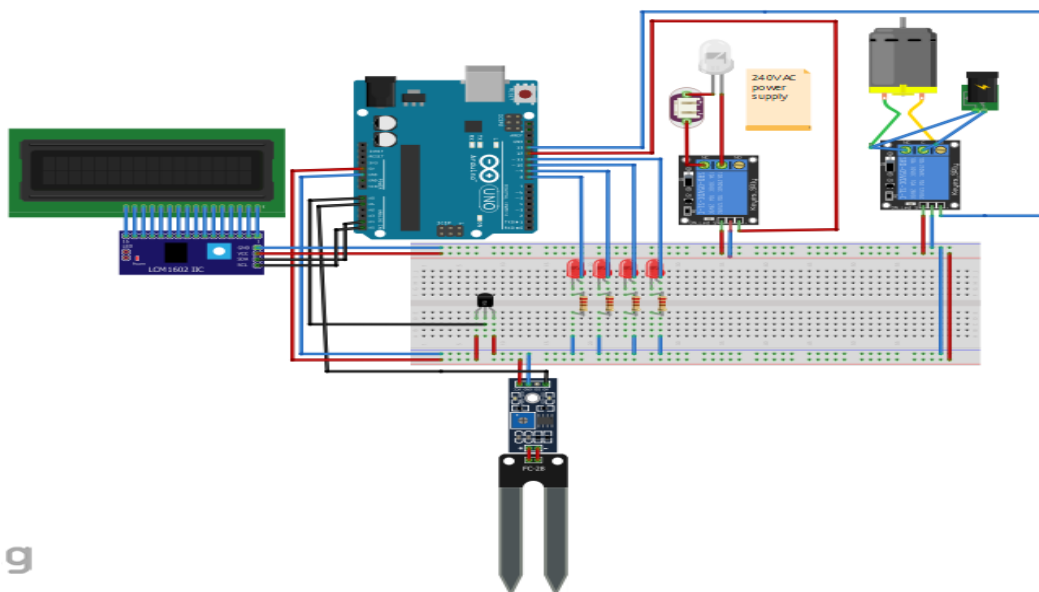
4. LƯU ĐỒ THUẬT TOÁN



Hình 15: Lưu đồ thuật toán

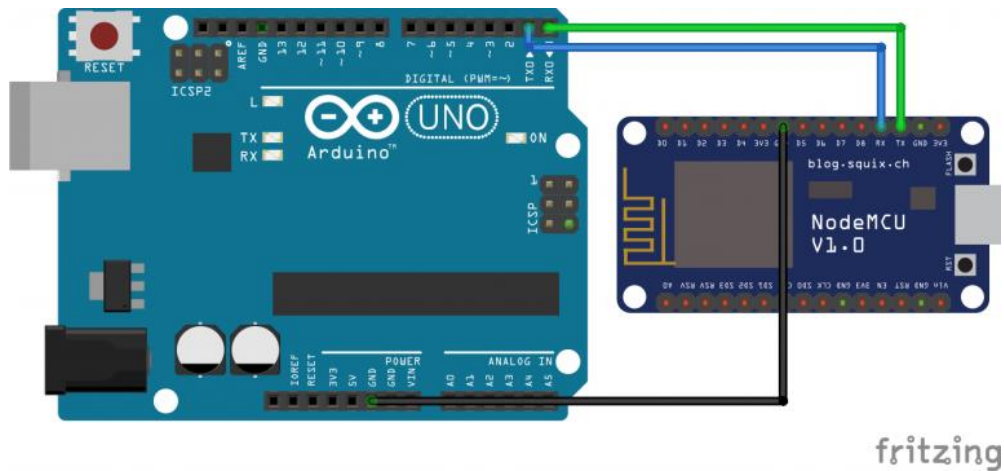
5.SƠ ĐỒ LẮP ĐẶT

5.1:KẾT NỐI GIỮA ARDUINO VÀ CÁC THIẾT BỊ.



Hình 16: Sơ đồ lắp đặt

5.2: KẾT NỐI GIỮA ARDUINO VÀ ESP8266.



Hình 17: Sơ đồ lắp đặt

Giao tiếp ESP8266 chuẩn UART

Có 2 cách để giao tiếp UART giữa ESP8266/NodeMCU:

1 là giao tiếp bằng cách code trên cả Uno/Mega và ESP8266/NodeMCU

2 là chỉ code trên Uno/Mega, trên ESP8266/NodeMCU flash thẳng firmware AT Command.

6 THIẾT KẾ APP ĐIỀU KHIỂN QUA ĐIỆN THOẠI.

6.1 Giới thiệu về Blynk

Blynk là một ứng dụng chạy trên nền tảng iOS và Android để điều khiển và giám sát thiết bị thông qua internet. Blynk không bị ràng buộc với những phần cứng cụ thể nào cả, thay vào đó, nó hỗ trợ phần cứng cho bạn lựa chọn như Arduino, Raspberry Pi, ESP8266 và nhiều module phần cứng phổ biến khác.

Những lý do nên sử dụng Blynk:

Dễ sử dụng: việc cài đặt ứng dụng và đăng ký tài khoản trên điện thoại rất đơn giản cho cả IOS và Android đơn giản cho cả IOS và Android

Chức năng phong phú: Blynk hỗ trợ rất nhiều chức năng với giao diện đẹp và thân thiện, bạn chỉ việc kéo thả đối tượng và sử dụng nó.

Không phải lập trình ứng dụng: nếu bạn không có kiến thức về lập trình app cho Android cũng như IOS thì Blynk là một ứng dụng tuyệt vời để giúp bạn khám phá thế giới IOTs.

Điều khiển, giám sát thiết bị ở bất kì đâu thông qua internet với khả năng đồng bộ hóa trạng thái và thiết bị.

Có ba thành phần chính trong nền tảng:

Blynk App - cho phép tạo giao diện cho sản phẩm của bạn bằng cách kéo thả các widget khác nhau mà nhà cung cấp đã thiết kế sẵn.

Blynk Server - chịu trách nhiệm xử lý dữ liệu trung tâm giữa điện thoại, máy tính bảng và phần cứng. Bạn có thể sử dụng Blynk Cloud của Blynk cung cấp hoặc tự tạo máy chủ Blynk riêng của bạn. Vì đây là mã nguồn mở, nên bạn có thể dễ dàng integrate vào các thiết bị và thậm chí có thể sử dụng Raspberry Pi làm server của bạn.

Library Blynk – support cho hầu hết tất cả các nền tảng phần cứng phổ biến - cho phép giao tiếp với máy chủ và xử lý tất cả các lệnh đến và đi.

Các tính năng:

Cung cấp API & giao diện người dùng tương tự cho tất cả các thiết bị và phần cứng được hỗ trợ

Kết nối với server bằng cách sử dụng: Wifi, Bluetooth, BLE Ethernet, USB(Serial) và GSM.

Các giao thức được sử dụng trong Blynk là giao thức MQTT,HTTP.

Các tiện ích trên giao diện được nhà cung cấp dễ sử dụng Thao tác kéo thả trực tiếp giao diện mà không cần viết mã

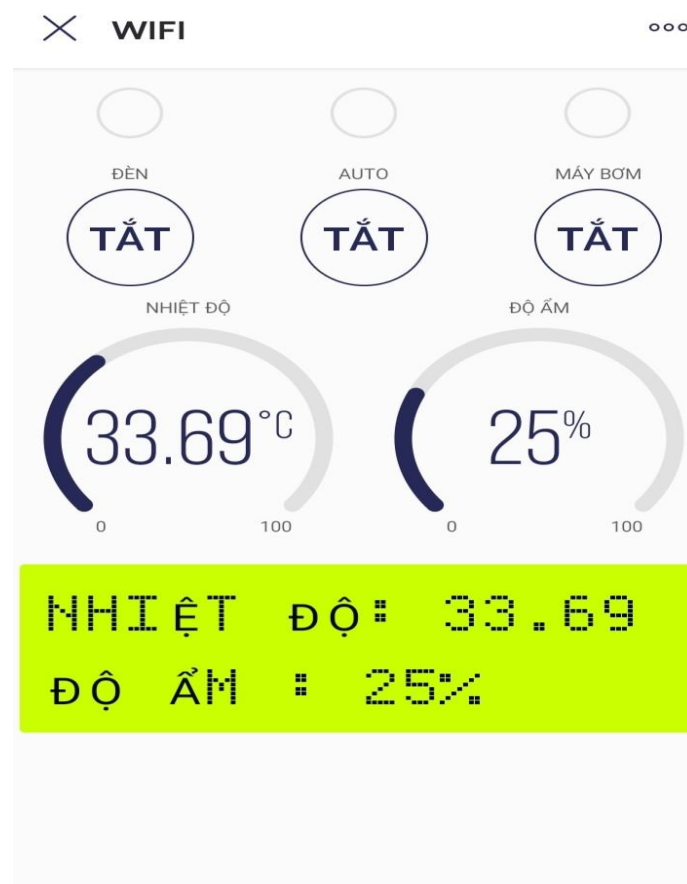
Dễ dàng tích hợp và thêm chức năng mới bằng cách sử dụng các cổng kết nối ảo được tích hợp trên blynk app

Theo dõi lịch sử dữ liệu.

Thông tin liên lạc từ thiết bị đến thiết bị bằng Widget Gửi email, tweet, thông báo realtime, v.v..... được cập nhật các tính năng liên tục!

6.2.Cấu hình và mô tả cách hoạt động của Blynk IoT.

6.2.1:Cấu hình của app Blynk IoT



Hình 18:Hình ảnh app blynk đã được cài đặt.

6.2.2:Cách hoạt động của blynk IoT app.

Như hình 15 đã cho thấy , cấu hình của một app IoT đã thiết kế có 3 nút bấm :

Nút bấm đầu tiên là AUTO , được hiểu là chia ra 2 trường hợp:

Trường hợp 1: Khi chuyển sang chế độ AUTO ở trạng thái TẮT, hệ thống sẽ ngừng hoạt động hoàn toàn. Điều này cho phép chúng ta điều khiển từ xa thông qua ứng dụng mà không cần phụ thuộc vào cảm biến hoặc chương trình hoạt động nào.

Trường hợp 2: Khi chuyển sang chế độ AUTO ở trạng thái BẬT, hệ thống sẽ được kích hoạt. Cả cảm biến, motor và đèn 220V có thể tự động bật và tắt dựa trên các thông số đã đo mà không cần sự can thiệp của con người.

Nút bấm thứ 2 là nút ĐÈN 220V. Khi nút này ở trạng thái tắt, đèn 220V sẽ tắt. Nếu muốn điều khiển bằng tay thông qua điện thoại, chỉ cần bật nó lên là đèn sẽ được bật.

Nút bấm thứ 3 là nút MÁY BƠM. Nó tương tự như nút ĐÈN 220V, cũng có 2 chế độ tắt và bật, cho phép điều khiển linh hoạt theo nhu cầu của người sử dụng.

Ngoài ra app còn có một màn hình LCD để hiển thị được nhiệt độ là bao nhiêu và độ ẩm là bao nhiêu để có thể dễ dàng quan sát hơn.

7.CHƯƠNG TRÌNH CHO ARNODEMCU

```
#include <LiquidCrystal_I2C.h> // Khai báo thư viện LCD I2C
#include <SoftwareSerial.h>
#include <SerialCommand.h>
LiquidCrystal_I2C lcd(0x27, 16, 2); // Khởi tạo đối tượng lcd với địa chỉ I2C của màn hình
int sensorPin = A1; // Chân đọc giá trị độ ẩm đất
int lm35Pin = A0; // Chân đọc giá trị nhiệt độ không khí

const int led1Pin = 8; // Chân điều khiển đèn LED 1
const int led2Pin = 9; // Chân điều khiển đèn LED 2
const int motorPin = 12; // Chân điều khiển motor
const int led3Pin = 10;
const int Den220 = 13; // chân điều khiển đèn 220v
const int LED220 = 11;
#define Rx 2
#define Tx 3
SoftwareSerial mySerial = SoftwareSerial(Rx,Tx);
String inputString = ""; // a String to hold incoming data
bool stringComplete = false; // whether the string is complete
String sendensp = "";
String data1 = "";
String data2 = "";
String data = "";
long last = 0;
int bienA,bienB;

void setup() {
  Serial.begin(9600);
  inputString.reserve(200);
  mySerial.begin(9600);

  lcd.init();
  lcd.backlight();
  pinMode(led1Pin, OUTPUT);
  pinMode(Den220, OUTPUT);
  pinMode(led3Pin, OUTPUT);
  pinMode(led2Pin, OUTPUT);
  pinMode(motorPin, OUTPUT);
  pinMode(LED220, OUTPUT);

  last = millis();
  Serial.print("GUI LAN DAU: ");
  Serial.println("C1");
  mySerial.println("C1");
}
```

```

void loop() {
    if(millis() - last >= 500){ //Hàm delay
        serialEvent();
        last = millis();
    }
}

void sendData(){
    if(bienA == 1)
    {
        data = data + "A1" + "D";
    }
    else if(bienA == 0){
        data = data + "A0" + "D";
    }
    if(bienB == 1)
    {
        data = data + "B1" + "D";
    }
    else if(bienB == 0){
        data = data + "B0" + "D";
    }
    mySerial.println(data);
}

//Hàm nhận dữ liệu từ ensp8266
void serialEvent() {
    while (mySerial.available()) {
        char inChar = (char)mySerial.read();
        inputString += inChar;
        if (inChar == '\n') {
            stringComplete = true;
        }
    }
    if (stringComplete) {
        Serial.print("DATA NHẬN ĐƯỢC: ");
        Serial.println(inputString);
        // clear the string:
        xulyONOFF(String(inputString));
        //
        inputString = "";
        stringComplete = false;
    }
}

void controllights() //Hàm xét điều kiện mặc định
{

```

```

float temperature = (5.0*analogRead(A0)*100.0/1024.0);
int lm35Value = analogRead(A0);
if (temperature < 30 )
{
    digitalWrite(Den220, HIGH);
    digitalWrite(LED220, HIGH);
    bienA = 1;
    lcd.setCursor(0, 1);
    lcd.print("Do C thap bat den");
    delay(1000);
}
if (temperature > 30)
{
    digitalWrite(Den220, LOW);
    digitalWrite(LED220, LOW);
    bienA = 0;
    lcd.setCursor(0, 1);
    lcd.print("Do C cao tat den");
    delay(1000);
}
}

void tudong() // Hàm chạy trạng thái tự động
{
    Serial.println("TÍNH TOÁN AUTO");
    int sensorValue = analogRead(sensorPin); // Đọc giá trị độ ẩm đất
    int lm35Value = analogRead(A0); // Đọc giá trị nhiệt độ không khí
    // Tính toán giá trị độ ẩm đất dưới dạng phần trăm
    int percentValue = map(sensorValue, 0, 1023, 100, 0);
    Serial.print("Do Am Dat: ");
    Serial.print(percentValue);
    data1 = String(percentValue);
    Serial.print("%\n");
    // Hiển thị giá trị nhiệt độ không khí
    float temperature = (5.0*analogRead(A0)*100.0/1024.0);
    Serial.print("Nhiệt độ không khí: ");
    Serial.print(temperature);
    data2 = String(temperature,2);
    Serial.println(" độ C ");
    data = "E" + data1 + "F" + data2 + "G";
    mySerial.println(data);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Do Am Dat: ");
    lcd.print(percentValue);
    lcd.print("%");
    lcd.setCursor(0, 1);
    // Chia ra 3 trường hợp

```



```

if (percentValue < 50)
{
    digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, LOW);
    digitalWrite(motorPin, HIGH);
    bienB = 1;

    Serial.print("Dat Thieu Do Am \n ");

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" Do Am: ");
    lcd.print(percentValue);
    lcd.print("%");
    lcd.setCursor(0, 1);
    lcd.print(" Bat Motor");
    delay(1000);
    lcd.setCursor(0, 1);
    lcd.print("Nhiet Do KK: ");
    lcd.print(temperature);
    lcd.print(" C");
    delay(1000);
    controllLights();
}
else if (percentValue >= 50 && percentValue < 59)
{
    digitalWrite(led1Pin, LOW);
    digitalWrite(led2Pin, HIGH);
    digitalWrite(led3Pin, LOW);
    digitalWrite(motorPin, LOW);
    bienB = 0;
    Serial.print("Do Am Du \n ");

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Do Am:");
    lcd.print(percentValue);
    lcd.print("%");
    delay(3000);

    lcd.setCursor(0, 0);
    lcd.print("Nhiet Do KK: ");
    lcd.print(temperature);
    lcd.print(" do C");
    delay(3000);

    lcd.setCursor(0, 1);
    lcd.print(" Tat Motor");
}

```

```

        delay(3000);
        controllLights();
    }
    else if (percentValue > 60)
    { // Đất dư độ ẩm
        digitalWrite(led2Pin, LOW);
        digitalWrite(led3Pin, HIGH);
        digitalWrite(motorPin, LOW);
        bienB = 0;
        Serial.print("Dat Du Do Am ");

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Do Am :");
        lcd.print(percentValue);
        lcd.print("%");

        lcd.setCursor(0, 1);
        lcd.print(" Tat Motor ");
        delay(1000);
        lcd.setCursor(0, 1);
        lcd.print("Nhiet Do KK: ");
        lcd.print(temperature);
        lcd.print(" do C");
        controllLights();
    }
}

// Hàm xử lý dữ liệu nhận từ ensp8266
void xulyONOFF(String inputString){
    if((inputString.indexOf("C1")>=0))
    {
        Serial.println("BẬT CHẾ ĐỘ AUTO");
        tudong();
        sendData();
        Serial.print("CHUỖI TỔNG:");
        Serial.println(data);
    }
    else if(inputString.indexOf("C0")>=0)
    {
        digitalWrite(led1Pin, LOW);
        digitalWrite(led2Pin, LOW);
        digitalWrite(led3Pin, LOW);
        digitalWrite(LED220, LOW);
        DATA();
        Serial.println("CHẾ ĐỘ THỦ CÔNG");
        if(inputString.indexOf("A1")>=0)
        {

```

```

        Serial.println("ON DEN 220V");
        digitalWrite(Den220, HIGH);
    }
    if(inputString.indexOf("A0")>=0)
    {
        Serial.println("OFF DEN 220V");
        digitalWrite(Den220, LOW);
    }
    if(inputString.indexOf("B1")>=0)
    {
        Serial.println("ON MÁY BƠM");
        digitalWrite(motorPin, HIGH);
    }
    if(inputString.indexOf("B0")>=0)
    {
        Serial.println("OFF MÁY BƠM");
        digitalWrite(motorPin, LOW);
    }
}

}

void DATA() //Hàm gửi dữ liệu nhiệt độ, độ ẩm
{
    Serial.println("TÍNH TOÁN MẶC ĐỊNH");
    int sensorValue = analogRead(sensorPin); // Đọc giá trị độ ẩm đất
    int lm35Value = analogRead(A0); // Đọc giá trị nhiệt độ không khí
    // Tính toán giá trị độ ẩm đất dưới dạng phần trăm
    int percentValue = map(sensorValue, 0, 1023, 100, 0);
    Serial.print("Do Am Dat: ");
    Serial.print(percentValue);
    data1 = String(percentValue);
    Serial.print("%\n");
    // Hiển thị giá trị nhiệt độ không khí
    float temperature = (5.0*analogRead(A0)*100.0/1024.0);
    Serial.print("Nhiệt độ không khí: ");
    Serial.print(temperature);
    data2 = String(temperature,2);
    Serial.println(" độ C ");
    if(percentValue < 10){
        data = "E" + data1 + "I" + "F" + data2 + "G";
    }else{
        data = "E" + data1 + "F" + data2 + "G";
    }
    mySerial.println(data);
    Serial.print("CHUỖI TỔNG:");
    Serial.println(data);
}

```

8.CHƯƠNG TRÌNH CHO ESP8266.

```
#include <Blynk.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <WiFiClientSecure.h>
#include <SoftwareSerial.h>
#include <SerialCommand.h>
#define Rx D6
#define Tx D5
SoftwareSerial mySerial(Rx, Tx);

#define BLYNK_TEMPLATE_ID "TMPL6cdPA3Ljw"
#define BLYNK_TEMPLATE_NAME "WIFI"
#define BLYNK_AUTH_TOKEN "pmOiXNS1uk7MxGwYOhAiC7LThx7wnB_G"
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "PTIT.HCM_SV"; // Thay thế bằng tên mạng Wi-Fi của bạn
char pass[] = ""; // Thay thế bằng mật khẩu Wi-Fi của bạn
unsigned long times=millis();
unsigned long timesDelay=millis();

BlynkTimer timer;

String inputString = ""; // a String to hold incoming data
bool stringComplete = false; // whether the string is complete
long last = 0;
float nhietdo,doam;
int bienA,bienB,bienC;
String sendAdruino = "";

void setup() {

  Serial.begin(9600);
  mySerial.begin(9600);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  inputString.reserve(200);
  last = millis();
}

void loop() {
  if(millis() - last >= 500)
  {
    senddata();
    serialEvent();
    last = millis();
  }
  Blynk.run();
}
```

```

    timer.run();
}

//Hàm đọc giá trị luồng dữ liệu từ app
BLYNK_WRITE(V0) {
    int pinValue = param.asInt();
    if (pinValue == 1) {
        bienA = 1;
    } else {
        bienA = 0;
    }
}

//Hàm đọc giá trị luồng dữ liệu từ app
BLYNK_WRITE(V1) {
    int pinValue = param.asInt();
    if (pinValue == 1) {
        bienB = 1;
    } else {
        bienB = 0;
    }
}

//Hàm đọc giá trị luồng dữ liệu từ app
BLYNK_WRITE(V2) {
    int pinValue = param.asInt();
    if (pinValue == 1) {
        bienC = 1;
    } else {
        bienC = 0;
    }
}

//Hàm gửi dữ liệu cho adruinno
void sendata(){
    sendAduino = "";
    if ( bienC == 1){
        sendAduino = sendAduino + "C1";
        Serial.print("Đã gửi đi: ");
        Serial.println(sendAduino);
        mySerial.println(sendAduino);
    }else if (bienC == 0){
        sendAduino = sendAduino + "C0";
        if ( bienA == 1){
            sendAduino = sendAduino + "A1";
        }else if(bienA == 0){
            sendAduino = sendAduino + "A0";
        }
        if ( bienB == 1){
            sendAduino = sendAduino + "B1";
        }else if(bienB == 0){

```

```

        sendAduino = sendAduino + "B0";
    }
    Serial.print("ĐÃ gửi đi: ");
    Serial.println(sendAduino);
    mySerial.println(sendAduino);
}
}
//Hàm đọc dữ liệu nhận được từ adruino
void serialEvent() {
    while (mySerial.available()) {
        // get the new byte:
        char inChar = (char)mySerial.read();
        // add it to the inputString:
        inputString += inChar;
        // if the incoming character is a newline, set a flag so the main loop can
        // do something about it:
        if (inChar == '\n') {
            stringComplete = true;
        }
        if (stringComplete) {
            Serial.print("ĐÃ NHẬN ĐƯỢC: ");
            Serial.println(inputString);
            xulyONOFF(String(inputString));
            inputString = "";
            stringComplete = false;
        }
    }
}
//Hàm xử lý dữ liệu nhận được từ adruino
void xulyONOFF(String inputString){
    if(inputString.indexOf("C1")>=0){
        Blynk.virtualWrite(V2,1);
        bienC = 1;
    }
    if(inputString.indexOf("A1")>=0){
        Blynk.virtualWrite(V0,1);
        bienA = 1;
    }
    else if(inputString.indexOf("A0")>=0){
        Blynk.virtualWrite(V0,0);
        bienA = 0;
    }
    if(inputString.indexOf("B1")>=0){
        Blynk.virtualWrite(V1,1);
        bienB = 1;
    }
    else if(inputString.indexOf("B0")>=0){
        Blynk.virtualWrite(V1,0);
    }
}

```

```

    bienB = 0;
}
if(inputString.indexOf("E")>=0){
    String DuLieuEF = "";
    if (inputString.indexOf("I")>=0)
    {
        DuLieuEF = inputString.substring(1,2);
        doam = DuLieuEF.toInt();
    }else{
        DuLieuEF = inputString.substring(1,3);
        doam = DuLieuEF.toInt();
    }
    DuLieuEF = inputString.substring(4,9);
    nhietdo = DuLieuEF.toFloat();
    Serial.print("ĐỘ ẨM: ");
    Serial.println(doam);
    Serial.print("NHIỆT ĐỘ: ");
    Serial.println(nhietdo);
    if(doam < 50)
    {
        Blynk.virtualWrite(V4, "THIẾU ĐỘ ẨM");
    }else{
        Blynk.virtualWrite(V4, "DƯ ĐỘ ẨM");
    }
    if(nhietdo > 30)
    {
        Blynk.virtualWrite(V3, "NHIỆT ĐỘ CAO");
    }else{
        Blynk.virtualWrite(V3, "NHIỆT ĐỘ THẤP");
    }
    Blynk.virtualWrite(V6, nhietdo);
    Blynk.virtualWrite(V7, doam);
}
}

```

8.1 Chú thích

Quy định:

*chuỗi gửi trạng thái máy bơm,đèn,auto hay thủ công

A1 A0: Biến đọc trạng thái đèn (0 là tắt,1 là bật)

B1 B0: Biến đọc trạng thái máy bơm (0 là tắt,1 là bật)

C1 C0: Biến đọc trạng thái auto-thủ công (0 là tắt,1 là bật)

*chuỗi gửi nhiệt độ và độ ẩm

E--F++,++G

-- là độ ẩm

++,++ là độ ẩm

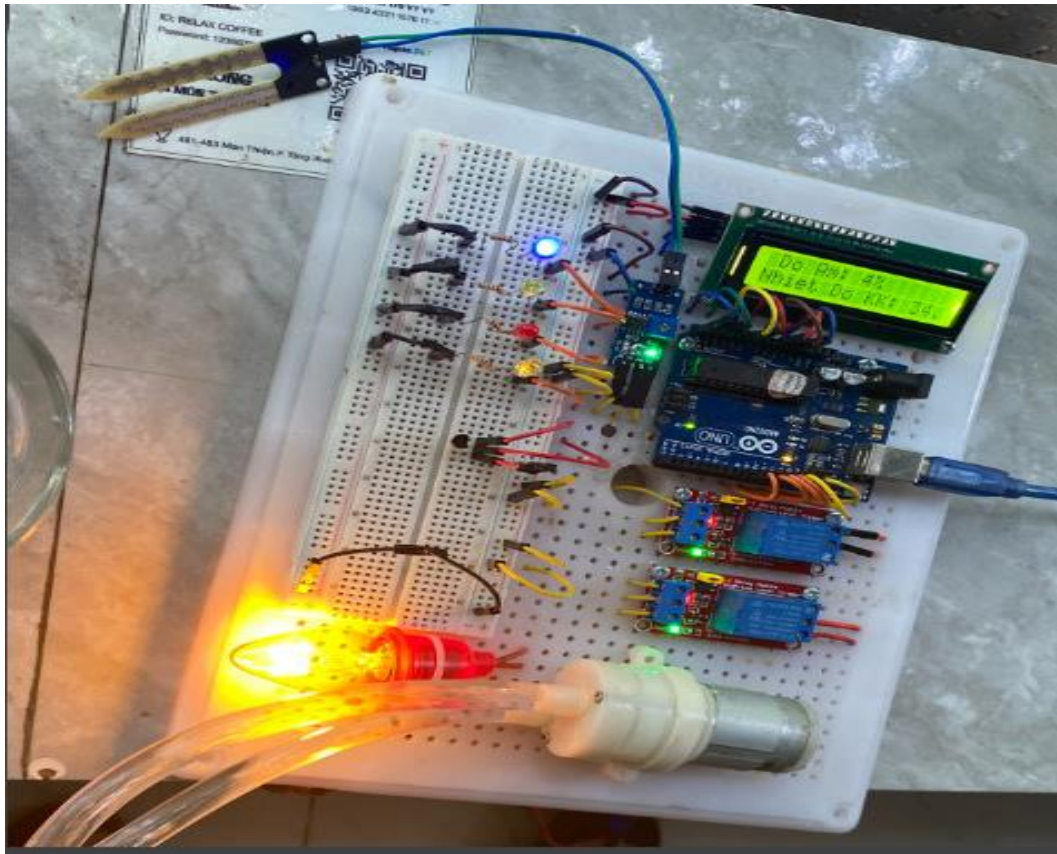
Dùng hàm substring để tách chuỗi

CHƯƠNG IV : KẾT QUẢ ĐẠT ĐƯỢC

1. Trường hợp 1

Khi cảm biến đo được độ ẩm đất có giá trị từ 0 % đến 50% là mức độ độ ẩm thấp, đất khô cần và không đủ nước để trồng cây phát triển. Vì vậy hệ thống sẽ tự động kích hoạt relay và máy bơm hoạt động. Đèn Led 1 màu xanh dương sẽ sáng biểu thị cho việc đất thiếu nước .

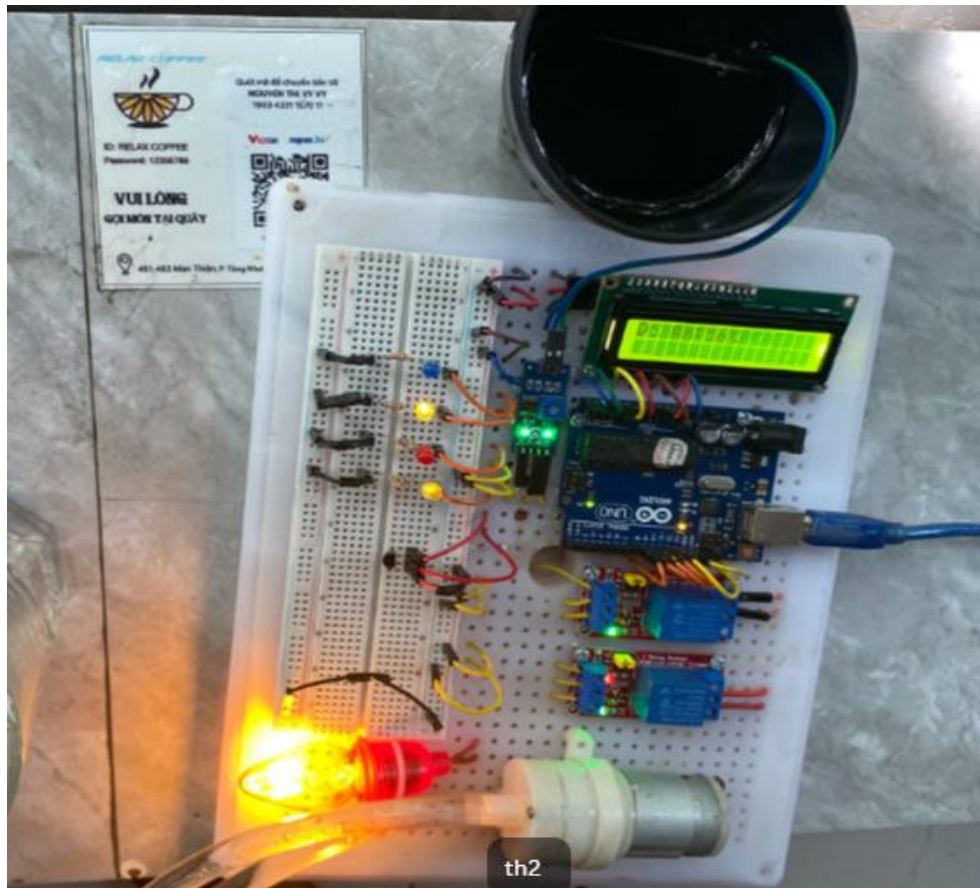
Khi cảm biến nhiệt (LM35) đo được thấp hơn 25 độ C, module điều khiển sẽ kích hoạt relay và bật đèn 220V để tạo ra ánh sáng và nhiệt độ cho cây trồng. Ngược lại khi nhiệt độ cao hơn 25 độ C, tắt đèn 220v . Đèn Led 4 màu vàng sáng biểu thị cho đèn 220 hoạt động



Hình 19: Trường hợp 1 thiếu độ ẩm motor hoạt động

2. Trường hợp 2

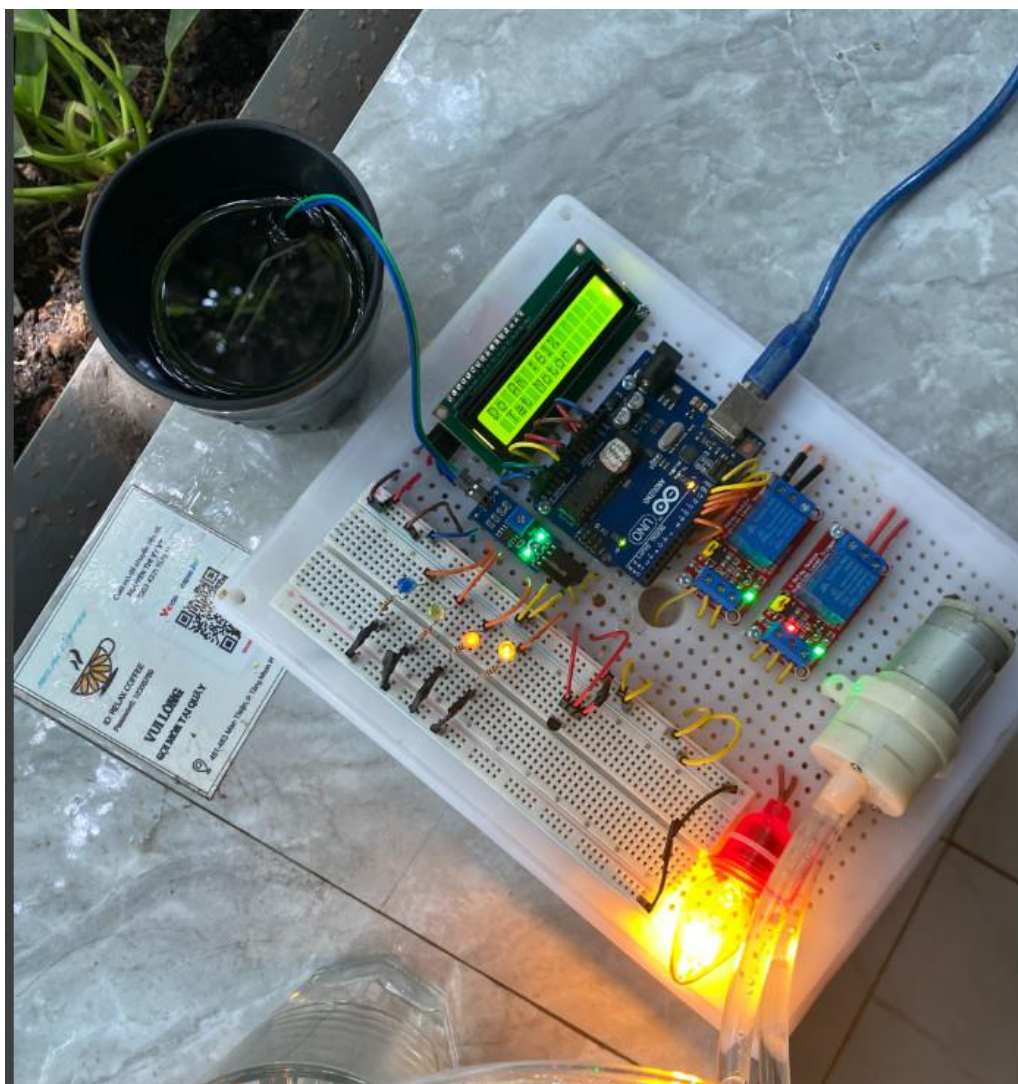
Khi cảm biến đo được độ ẩm từ 50 đến 70% là mức độ ẩm đủ , đất đủ độ ẩm không cần tưới thêm nước , vì vậy tắt relay để tắt motor .Đèn Led 2 màu vàng sẽ sáng biểu hiện cho đất đủ độ ẩm.



Hình 20: Trường hợp 2 đủ độ ẩm motor tắt đèn 220v hoạt động

3. Trường hợp 3

Khi cảm biến đo được độ ẩm lớn hơn 70% là mức độ ẩm dư, đất dư độ ẩm không cần tưới thêm nước, vì vậy tắt relay để tắt motor. Đèn Led 3 màu đỏ sẽ sáng biểu hiện cho đất dư độ ẩm.



Hình 21: Trường hợp 3 dư độ ẩm motor tắt đèn 220v sáng

CHƯƠNG V : ĐỀ XUẤT PHÁT TRIỂN ĐỀ TÀI

Với đề tài này nếu có đủ thời gian nghiên cứu thì có thể mở rộng ra ngoài việc sử dụng cảm biến độ ẩm đất , nhiệt độ còn có thể sử dụng thêm các loại cảm biến khác như: cảm biến tốc độ động cơ để điều chỉnh lượng nước tưới tiết kiệm hiệu quả, cảm biến đo độ pH để đo độ pH trong nước tưới, từ đó kịp thời xử lý nếu nước bị nhiễm phèn.

Có thể tạo ứng dụng để điều khiển hệ thống tưới nước tự động từ xa bằng Blynk. Blynk là một nền tảng đám mây IoT cho phép bạn tạo các ứng dụng IoT và điều khiển các thiết bị thông minh từ xa thông qua các thiết bị di động của bạn.

TÀI LIỆU THAM KHẢO

[1] Lê Trường Thuận , **Đồ án điện công nghiệp: Thiết kế hệ thống tưới cây tự động sử dụng Arduino**,<https://tailieu.vn/doc/do-an-dien-cong-nghiep-thiet-ke-he-thong-tuoi-cay-tu-dong-su-dung-arduino-2271944.html>

[2]. Green Technology , **Hệ thống hẹn giờ tưới cây theo thời gian thực sử dụng Arduino + DS3231**,
<https://www.youtube.com/watch?v=-FikO9vrqCY&t=268s>

[3]Nguyễn Thủy , **Cấu tạo của hệ thống tưới phun sương tự động**,<https://tuoinongnghiep.net/cau-tao-cua-thong-tuoi-phun-mua-tu-dong/>

[4]Link github source, https://github.com/Nadh2413/Nhom1_PhatTrieHeThongVaUngDungIoT.git